

第一章:

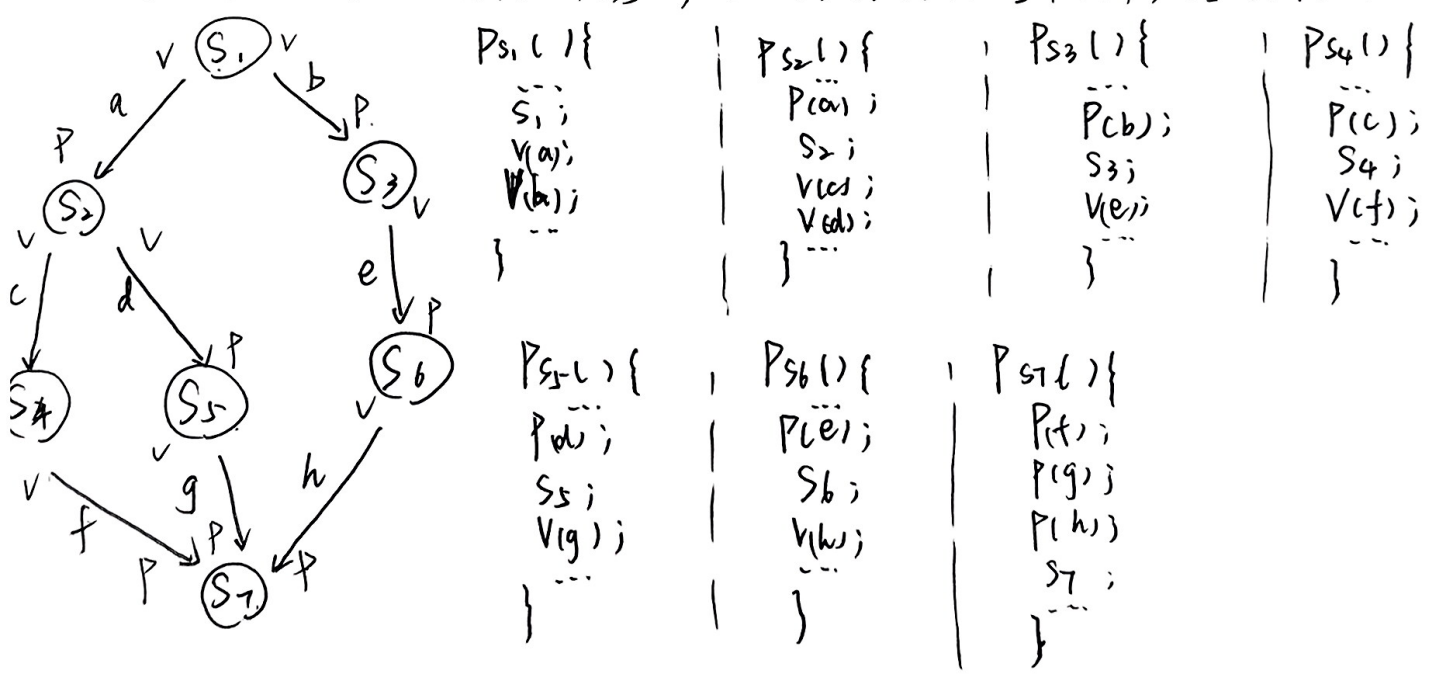
- 一、1. 操作系统: 是管理计算机软硬件资源的系统软件.
- 4. 实时操作系统: 可以及时, 按照时间限制内完成任务的操作系统.
- 5. 互斥共享: 一段时期内, 在某一进程使用某一资源时, 其余资源无法使用该 P. 资源, 直到这一进程使用完毕.
- 三、1. 软件, 硬件系统
- 3. 硬件资源, 软件资源
- 5. 共享性, 异步性.

第二章:

- 一、3. 临界区: 访问临界资源的那一段代码
- 4. 进程同步: 在异步环境下, 协调多个进程的执行顺序, 使其有序的进行.

- 三、1. 动态性, 异步性, 共享性.
- 5. 动态, 静态.
- 6. 互斥.
- 8. 共享存储器, 管道, 消息传递.
- 9. 就绪, 运行, 阻塞.

五: 2. $S_1 \rightarrow S_2, S_2 \rightarrow S_4, S_4 \rightarrow S_5, S_1 \rightarrow S_3, S_3 \rightarrow S_6, S_4 \rightarrow S_7, S_5 \rightarrow S_7, S_6 \rightarrow S_7$.



3. ~~互斥~~ 互斥：对共享资源的访问互斥进行。

同步 ① 互斥为互斥时所以放入水果，

② 父亲放入苹果女儿才可取。

③ --- 橘子儿子 ---

S mutex = 1;
S empty = 5;
S apple = 0;
S orange = 0;

Father() {
while(1) { // 放入水果;
P(empty);
P(mutex);
if (fruit == apple)
V(apple);
else V(orange);
V(mutex);
}
}

daughter() {
while(1) {
apple
P(~~orange~~);
P(mutex);
取苹果;
V(mutex);
V(~~empty~~);
}
}

Son() {
while(1) {
P(orange);
P(mutex);
取橘子;
V(mutex);
V(empty);
}
}

4. 同步关系：1为空时 → PA可写入1

1有值，2为空时 → PB可将1复制到2。

2有值 → PC可打印。

S empty1 = 1, empty2 = 1;

S full = 0, full2 = 0;

PA() {
while(1) {
...
P(empty1);
V(full1);
}
}

PB() {
while(1) {
...
P(full1);
P(empty2);
V(empty1);
V(full2);
}
}

PC() {
while(1) {
...
P(~~empty~~ full2);
V(empty2);
}
}

5. S customer = 0, wait = 0, barber = 0, pay = 0; S mutex = 1;

barber() {
while(1) {
P(customer);
P(mutex);
wait --;
V(barber);
V(mutex);
P(pay);
}
}

customer() {
while(1) {
P(mutex);
if (wait > 0) {
wait ++;
V(customer);
V(mutex);
P(barber);
V(pay);
else V(mutex);
}
}
}

第三章

1. 2. 是对处理机进行分配.

3. 周转时间与从作业被接受给系统开始到作业完成为止的这段时间间隔.

4. 死锁: 无法获得所需资源而无法放下手中已有资源 形成的僵持状态.

三. 2. 提交, 就绪, 终止.

6. 预防死锁, 避免死锁, 死锁的检测与解除, 死锁

五:

1.

FCFS:

	0	2	4	6	8	10	12
	A	B	C	D	E		
FCFS:	进程	到达t	服务t	完成t	周转t	带权周转t	
	A	0	3	3	3	1	
	B	2	6	9	7	1.17	
	C	4	4	13	9	2.25	
	D	6	5	18	12	2.4	
	E	8	2	20	12	6	

$$\text{平均周转t: } (3+7+9+12+12) \times \frac{1}{5} = 8.6$$

$$\text{平均带权周转t: } (1+1.17+2.25+2.4+6) \times \frac{1}{5} = 2.564$$

SJF: 进程
(非抢占).

	0	2	4	6	8	10	12
	A	B	C	D	E		
	A	0	3	3	3	1	
	B	2	6	9	7	1.17	
	C	4	4	15	11	2.75	
	D	6	5	20	14	2.8	
	E	8	2	11	3	1.5	

$$\text{平均周转t: } (3+7+11+14+3) \times \frac{1}{5} = 7.6$$

$$\text{平均带权周转t: } (1+1.17+2.75+2.8+1.5) \times \frac{1}{5} = 1.844$$

SJF	A	0	3	3	3	1
抢占	B	2	6	15	13	2.7
	C	4	4	8	4	1
	D	6	5	20	14	2.8
	E	8	2	10	2	1

平均周转时间: $(3+13+4+14+2) \times \frac{1}{5} = 7.2$

平均带权周转时间: $1+2.7+1+2.8+1 = 1.594$

HRRN:	A	0	3	3	3	1
	B	2	6	9	7	1.17 1.17
	C	4	4	13	9	2.25
	D	6	5	18	12	2.4
	E	8	2	20	12	6

平均周转时间: $(3+7+9+12+12) \times \frac{1}{5} = 8.6$

平均带权周转时间: $(1+1.17+2.25+2.4+6) \times \frac{1}{5} = 2.564$

RR	A	0	3	4	4	1.34
(q=1)	B	2	6	18	16	2.67
	C	4	4	17	13	3.25
	D	6	5	20	14	2.8
	E	8	2	15	7	3.5

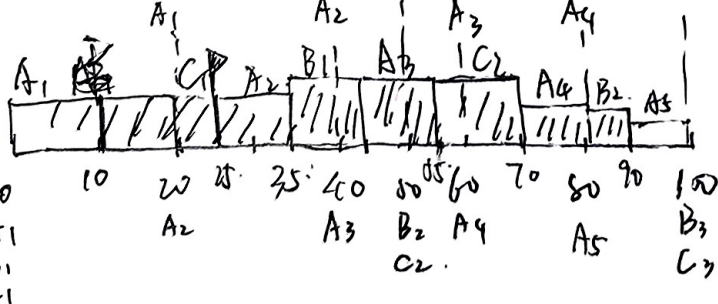
平均周转时间: $(4+16+13+(4+7)) \times \frac{1}{5} = 10.8$

平均带权周转时间: $(1.34+2.67+3.25+2.8+3.5) \times \frac{1}{5} = 2.712$

2. A. 每 20. 10.

B 50 10

C 50 B1 C1 15.



OS: $A_1: 20-10=10$

$B_1: 50-10=40$

$C_1: 50-15=35$

~~$A_2: 40-10=30$~~

~~$B_2: 70-10=60$~~

10S: $B_1: 50-10-10=30$

$C_1: 50-15-10=25$

25S: $B_1: 50-10-25=15$

$A_2: 40-10-25=5$

45S: $B_2: 100-10-55=35$

$C_2: 100-15-55=30$

3、 ~~Need~~ Allocation = Max - Need.

16 22.

(1) Progress.	Allocation.			
P ₀	0	0	3	2
P ₁	1	0	0	0
P ₂	1	3	5	4
P ₃	0	3	3	2
P ₄	0	0	1	4

$P_0 \rightarrow P_3 \rightarrow P_1 \rightarrow P_2 \rightarrow P_4$.

存在安全序列 $\{P_0, P_3, P_1, P_2, P_4\}$

∴ 该状态安全.

(2) Request \leq Need_{P₂} ; Request \leq Available

先假设分配给 P₂.

Available = (0, 4, 0, 0).

Allocation_{P₂} = (2, 5, 7, 6).

Need_{P₂} = (1, 1, 3, 4)

无安全序列. Available 不满足任何请求, 进入不安全状态.

故不可分配.

(3) 并不会, 因为此时无其余进程申请新的资源, 并在阻塞状态, 只有不提出新的请求而导致各进程因得不到资源而阻塞并可能形成循环等待链时, 才会进入死锁.